# SeriesNet: A Dilated Causal Convolutional Neural Network for Forecasting

**Krist Papadopoulos**[*]
University of Toronto

April 2018

**Abstract**

In this paper, state of the art deep learning techniques for time series forecasting were surveyed and a dilated causal convolutional neural network was developed (i.e. SeriesNet) based on the WaveNet architecture to forecast time series. It was found that SeriesNet without data preprocessing and ensemble methods achieved comparable results to top performing deep learning models on the international Computational Intelligence in Forecasting (CIF2016) competition dataset. This work extends previous work on using convolutional neural networks for time series forecasting.

## 1 Introduction

The ability to forecast trends from data is an important application area of statistics to many domains of business, science and engineering. Classical time series forecasting includes a number of techniques to create parameterized models of different types of stochastic processes [1]. These processes tend to be difficult to forecast due to features such as: non-linearity, non-stationarity and unknown dependencies. Various machine learning (ML) techniques for sequential data prediction were reviewed in [2] and evaluated in [3] for different time-series, demonstrating that an artificial neural net (ANN) with one hidden layer of non-linear nodes produced the highest forecasting accuracy. The investigations in [2] and [3] did not consider deep learning (i.e. ANN with more than one hidden layer). The advantage of deep learning is that multiple layers of hidden non-linear nodes in a ANN learn hierarchical features of the input data which when combined with ANNs ability to approximate any non-linear function between the input and output data, creates models with significant expressive power [4]. This ability of deep ANNs led to significant advances in automatic speech recognition [5] and image classification [6][7].

A survey of recent deep learning techniques for time-series prediction was provided in [8]. A deep ANN with feature preprocessing was successful in predicting various time-series more accurately than classical techniques [9] and the expressive power of deep ANNs in approximating non-linear functions over time was demonstrated in estimating chaotic dynamics [10], estimating non-linear system operation such as Google data center power usage effectiveness [11] and robotic control [12]. Deep ANNs ability to approximate sequential data may not be robust to generalize future predictions because the deep ANN architecture does not take into account correlated sequential inputs or feedback from prediction outputs. This may be important in shorter duration or sparse time-series where sufficient data is not available to learn the underlying structure and not incorporating sequential structure into deep ANNs for time-series may exacerbate concerns of deep ANNs overfitting to training data while not generalizing to unseen data [13]. These concerns have limited the use of deep ANNs for

[*] M.A.Sc. student in Electrical and Computer Engineering

time-series prediction although further investigations demonstrated that a single layer ANN with feature pre-processing was effective in predicting shorter duration time-series [14] and that a deep ANN could predict company financial fundamentals over a long-term training data input window without overfitting [15].

The incorporation of innovative structures into deep ANNs for sequential data prediction tasks resulted in significant improvements in language translation [16] with long short term memory recurrent neural networks (LSTM-RNN) and text-to-speech [17] with dilated causal convolutional neural networks (DC-CNN). The LSTM-RNN was applied to time-series forecasting [18], achieving the best performance at the CIF2016 forecasting competition [19]. To leverage learning from multiple related time-series, the LSTM-RNN was extended to forecasting large-scale multivariate time-series [20][21][22]. CNN designs were applied to time-series prediction in [23][24]. The DC-CNN was demonstrated to have better performance than a fully connected CNN design [25] and comparable performance to a LSTM-RNN on financial time-series [26]. CNN designs have also been extended to multivariate time-series [25][26][27][28].

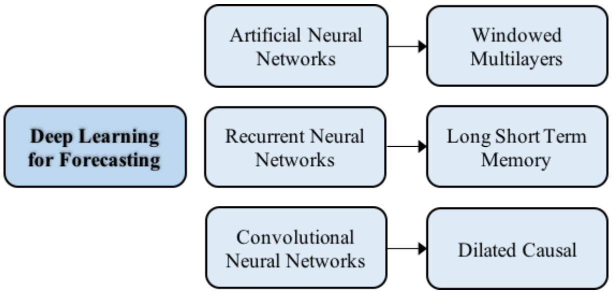
Deep Learning for Forecasting
Artificial Neural Networks
Windowed Multilayers
Recurrent Neural Networks
Long Short Term Memory
Convolutional Neural Networks
Dilated Causal

Figure 1: Outline of the survey areas covered in this paper

## 2 Approach

The purpose of this paper is to extend the survey of deep learning techniques for time-series in [8] and extend the empirical investigations of DC-CNN time-series forecasting in [25][26]. The recent work in time-series forecasting with deep ANNs, LSTM-RNNs and DC-CNN models for univariate and multivariate forecasting was reviewed in Section 2. In Section 3, a DC-CNN model (i.e. SeriesNet) was developed based on the WaveNet architecture [17]. The experiments of SeriesNet on the Computational Intelligence in Forecasting (CIF2016) competition dataset were presented in Section 4.

The DC-CNN model architecture was selected for further examination in this paper because of the following characteristics: demonstrated better performance to other CNN designs for time-series prediction [25], simpler network architecture and less model training time compared to LSTM-RNN, and potentially less data preprocessing and number of observations required for accurate forecasting compared to the LSTM-RNN. The hypothesis is that a DC-CNN model architecture based on WaveNet will exhibit performance similar to the LSTM-RNN submissions on the CIF2016 dataset due to the ability of the convolutional layers in learning local structures throughout the time series that can be effectively generalized over the prediction horizon by the autoregressive structure of the WaveNet architecture. Other types of datasets were not tested but the CIF2016 datasets consists of 72 time series with a mixture of real versus artificially generated series of different lengths and prediction horizons. Another limitation of this paper was that only deep learning techniques for time series forecasting were reviewed. Advances in other forecasting techniques such as Bayesian estimation, Gaussian process or ensemble methods were not included. The limitations of this paper were discussed in Section 5.

# 3 Related Work

In [8], deep learning techniques for time series forecasting were reviewed. This survey reviewed methodologies such as hybrid ANN models, Deep Belief Networks, Stacked Autoencoders and fully connected CNNs. In this paper, the review of deep ANNs in time series forecasting will be continued to highlight results not covered in [8]. The other methodologies covered in [8] were not reviewed further. The survey in [8] was extended in this paper by reviewing different results with deep ANNs and reviewing recent techniques that utilize LSTM-RNNs and DC-CNNs for time series forecasting.

## 3.1 Deep Artificial Neural Networks

The advantage of using a neural network as opposed to traditional autoregressive techniques for forecasting is its ability to approximate non-linear functions. The concern with using neural networks is that they may overfit to the input data and therefore not generalize especially in the presence of noisy data. An approach taken for forecasting with neural networks was to develop a staggered approach to avoid overfitting by first approximating the input linearly then adding hidden layers in the neural network if required to represent non-linear portions of the input [9]. This modelling strategy may help prevent overfitting for forecasting but it would depend on the nature of the data in the sliding window or lag structure defined to capture the portion of the time series input that would be fed to the neural network. This required different tests in [9] to determine the most effective window for the model and data. In [10], Frank et. al. examined the sliding window approach of using the different lags of a time series as inputs to the neural network to predict the next time step. They investigated the size of the sliding window required to correctly capture the dynamics of the system to permit the neural network to forecast the pattern accurately. Using different techniques, the sliding window size or embedding dimension was determined for the Lorenz attractor and it was used to successful forecast the pattern with the neural network. Although in [10], the application of different embedding dimensions to forecast a real dataset was not as successful. If the time series pattern such as the Lorentz attractor example is sufficiently represented within the defined sliding window, then the deep neural network may be able to approximate the function accurately.

This was demonstrated for forecasting data centre power usage in [11] where Rao utilized two years of data centre power usage to train a deep ANN to accurately forecast future power usage. No sliding windows were used as all the data from the two years was used for training and testing. The pattern of data centre power usage was non-linear although it exhibited a periodic pattern over the period selected that the neural network was able to approximate.

Crone and Häger [14], implemented a shallow neural network with only one hidden layer of up to 20 nodes with varying lag structures at each time step to model input data trend and seasonality. Different support points were selected in the lag structure (e.g. t-12, t-24 where annual seasonality might be a factor) for training the neural network. Double support points were experimented with (e.g. t-12, t-13, t-24, t-35) but it was found that the single support points over 4 lags (i.e. t-1, t-12, t-24, t-35) were the most effective in forecasting the input data from the CIF2016 dataset. In [15], it was similarly found by Alberg and Lipton that using a deeper neural network with a yearly lag structure up to 4 year prior at each month on a training set of historic company public financial data produced predictions better than traditional forecasting techniques and comparable to RNN. The disadvantage of the lag structure techniques in [14] and [15] were that hundreds of neural networks models were created to implement the lag structure across the different time steps in the training dataset.

## 3.2 Long Short Term Memory Recurrent Neural Networks

RNNs, through the LSTM-RNN implementation have demonstrated success in sequential prediction tasks such as language translation [16]. The advantage of the LSTM-RNN architecture over the standard neural network is the ability to capture memory of past states and modify this with recent states in one model. In the neural network approach for forecasting, information outside of the current sliding window used for the input does not contribute to the forecast where in the case of an LSTM model previous forecast information is incorporated. In [18], Smyl and Kuber tested the performance of LSTM models on the M3 competition (2000) yearly datasets. The first model was a baseline LSTM model with a sliding window of 7 because the shortest time series had 14 values and a log transformation of the input data was applied to reduce to the scale of the inputs. The same LSTM model with features extracted from exponential smoothing of the input data was also tested. It was shown that on the 645 yearly time series with input length of 14 to 40 values and maximum prediction horizon of 6, that the baseline LSTM preformed equivalent to the top submission on the yearly dataset which did not incorporate deep learning techniques. The LSTM model with features developed from exponential smoothing of the input produced lower error than the same top submission on the yearly dataset.

Further work by Smyl demonstrated that an LSTM model with input data preprocessing produced the lowest forecast error in the CIF2016 competition [19]. The baseline LSTM model developed by Smyl incorporated an ensemble of two LSTM models to account for the two forecast horizons of 6 and 12 used in the CIF2016 competition. Based on the forecast horizon, a different sliding window was prepared for the corresponding time series. The baseline LSTM model finished 10th overall in the CIF2016 competition. This model was augmented by Smyl to produce the winning model (i.e. LSTM-DES) with the lowest overall forecast error [19]. It included log transformation of the inputs and seasonal decomposition by Loess resulting in vectors representing trend, season and reminder. A sum of trend and reminder was input to the LSTM model for training and the seasonality component was added after. A more detailed description of the preprocessing method was provided in [21].

Current developments in LSTM forecasting extended the modelling approach described above with non-linear encoding of the input data and multivariate time series analysis. As seen from the CIF2016 competition results and reported by Laptev et. al. in [22], baseline LSTM models do not necessarily perform well on time series forecasting without different types of data preprocessing. The difficulty with the data processing techniques described above was that they might not be applicable on other datasets or over certain forecast horizons. The approaches taken by Flunkert et. al. [20] and Laptev et. al. [22], were to apply a non-linear embedding such as an encoder/decoder or an autoencoder to the input time series which included multivariate inputs to automatically capture non-linear features from the datasets. The transformed input was provided to an LSTM model to produce the forecasts. It was found that the non-linear embedding of the input and model training with multivariate inputs improved the forecast accuracy. These techniques formed the basis of production forecast systems at Amazon [20] and Uber [22].

## 3.3 Dilated Causal Convolutional Neural Networks

CNNs have demonstrated state of the art results in image classification tasks [6][7] by learning filters that represent different features through the hierarchy of convolutional layers in the network. The motivation of applying the CNN architecture to time series forecasting was to exploit the filter feature extraction and composition ability demonstrated during image classification and CNNs are easier to train than RNNs because of the implementation of the convolution operation as opposed to recursion. Like ANNs, CNNs could be trained using a sliding window approach for time series forecasting, although this would only produce a forecast per window and would not take advantage of the CNN ability to learn local features across different filters in the convolutional network. An initial attempt was to apply a causal 1D convolution in fully CNNs [23]. The causal convolution was necessary to preserve the causal ordering of the input time series. This model was not deep as it only utilized 5 layers with filter sizes that were upsampled through the layers to preserve the size of the input going through the layers. The model was referred to as an undecimated fully CNN and demonstrated comparable results to RNN and LSTM-RNNs on the datasets tested.

To extend the depth of the causal 1D convolution, Borovykh et. al. [25] introduced dilated causal convolutions to time series forecasting based on the WaveNet architecture for audio waveforms [17]. Similar to WaveNet, layers of dilated causal convolutions with a filter size of 2 were stacked to create a deeper receptive field than possible for the same number of layers in a fully CNN. The dilated causal CNN preserved the input size to the output while capturing longer term dependencies in the input time series without the filter size and number of filters required by the fully CNN models. It was shown in [25] that the performance of a dilated causal CNN (i.e. Augmented WaveNet) exceeded the performance of a fully CNN and in [26], Borovykh et. al. demonstrated that the Augmented WaveNet outperformed an LSTM implementation on the dataset tested. The details of the dilated causal CNN architecture of the Augmented WaveNet [26] and the architecture implemented in this paper are described in section 3. The experiments performed in this paper with a dilated causal convolution implementation are described in Section 4.

As described for LSTM models above, multivariate extensions to CNNs for time series forecasting have also been investigated. In the Augmented WaveNet model [26], an option was developed to condition the input on one related time series to create a bi-variate forecast. It was shown that including a related time series into the forecast improved the results especially for time series that had long term correlations. An approach developed by Yi et. al. in [27] was to group related time series by correlation through clustering techniques to exploit latent features from multiple time series in a CNN. Further developments on multivariate forecasting from CNNs involved creating forecasts based on a weighted sum of regressors learned from the input time series [28]. The model called a Significance - Offset CNN used a non-linear gate to weigh outputs of the CNN layers which represent local dependencies independent on the relative time position with candidate predictors based on the inputs through a 1X1 convolution that are independent of time position.

## 4 Network Architecture

The structure of a dilated causal convolution is presented in Figure 2. The causal convolution preserves the time sequence ordering of the input such that the predictions from the model do not depend on future time steps. The dilated convolution in each convolutional layer skips input values with a certain step size. This extends the receptive field of the network exponentially without requiring many convolutional layers or large filter sizes [17]. For example, in Figure 1, the network with 4 dilated convolutional layers, dilation factor of 2 and filter size of 2 has a receptive field of 16 which indicates that 16 inputs influence the output. The expression for the receptive field (r), with the number of layers L and filter size k is given by (1) below.

$$r = 2^{L-1}k \tag{1}$$

The WaveNet architecture shown in Figure 3 utilized multiple layers of dilated causal convolutions along with model features such as gated convolutions [29] using Sigmoid and Tanh activation functions, residual [30] and parameterized skip connections that led to successful modelling of audio input with the focus of generating a probability distribution of the learned

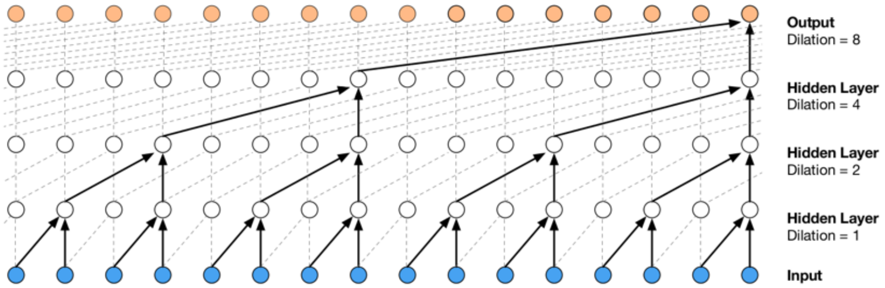

Figure 2: Dilated causal convolutional neural network structure

audio samples to be sampled at prediction time. The features of the WaveNet architecture were reviewed for time series forecasting and an architecture called Augmented WaveNet was developed [25][26]. Augmented WaveNet similarly used multiple dilated causal convolutional layers as shown in Figure 4. It was found that the gated convolution used in WaveNet was not effective for time series inputs and therefore the rectified linear unit (ReLU) [6] activation function was used between the layers. The Augmented WaveNet also used residual connections between the input and the output of the convolutions as in the WaveNet model to improve the training of the network. Although in the Augmented WaveNet, parameterized skip connections from each dilated convolutional layer to an output layer were not used. The forecast from the Augmented WaveNet was produced directly from the last stacked layer L, passed through a 1X1 convolution to obtain the output as shown in Figure 4.

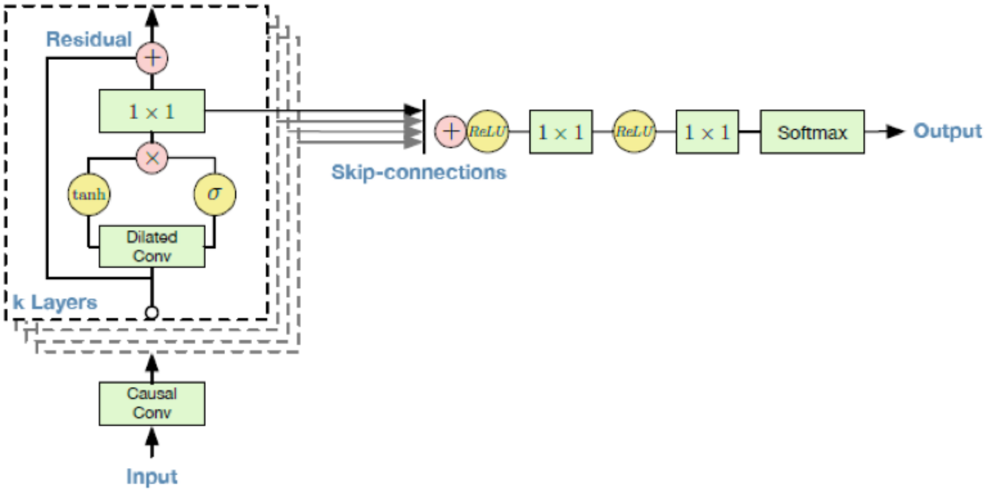

Figure 3: WaveNet architecture

A dilated causal convolutional neural network (i.e. SeriesNet) was developed based on the WaveNet architecture in [17] and the Augmented WaveNet in [26] to forecast time series. The SeriesNet architecture is shown is Figure 5. 7 dilated causal convolution layers with 32 filters were stacked with a dilation of 2 and filter size of 2 for each layer resulting in a total receptive field of 128. Since the longest series in the CIF2016 dataset was 108 inputs, a longer receptive field was used to capture longer term trends. It was found that 32 filters in each layer balanced underfitting and overfitting of the input data.

The gated convolution used in WaveNet after the dilated convolutional layers was tested along with the ReLU activation used after the dilated convolutional layers in the Aug-

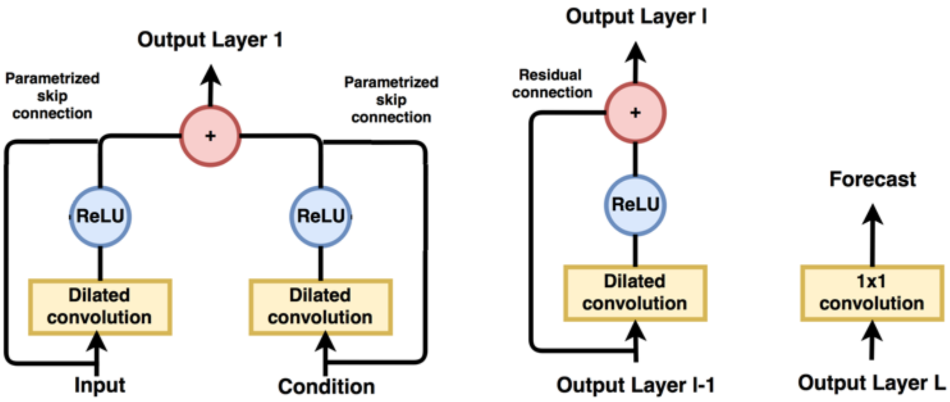

Figure 4: Augmented WaveNet architecture

mented WaveNet. It was found that the gated convolution was not effective in converging to low training error across all time series tested. The ReLU activation performed more consistently in reducing the training error across all time series. A concern with using the ReLU activation for time series forecasting was that the hidden representations created by the dilated convolutional layers with ReLU activation may be limited due to the ReLU only passing positive values. Also the activation of the ReLU is not mean centered because it is non-negative. Therefore, a bias is introduced into the next layer which is not desirable for developing the representation of the time series in the dilated convolutional layers. Batch normalization [31] was tested to normalize the activation of the ReLU although it was found that for some time series it produced improved results but for other series it resulted in the training not converging. The LeakyReLU [32] and the scaled exponential linear unit (SELU) [33] were experimented with to determine whether they could achieve better performance than the ReLU by removing bias from the activation. It was found that using the SELU activation in the dilated convolutional layers achieved efficient, stable training and resulted in the best performance. The self-normalizing properties of the SELU shown in [33] may have led to more robust representations of the time series being learnt by the hidden layers. The Tanh activation function was also investigated on its own in the dilated convolutional layers. Its performance was the worst of all activation functions tested likely because for the different time series inputs, the activations from the Tanh were saturated which prevented learning. Preprocessing the time series inputs into a different scale may have permitted the Tanh activation to perform more effectively but this was not tested.

The output from SeriesNet was formed from parameterized skip connections from each dilated convolutional layer. The outputs were summed and passed through a ReLU activation before being output from a 1X1 convolution. This structure was similar to the WaveNet architecture. The forecasts from this configuration were similar to forecasts produced from Augmented WaveNet configuration that was tested. Although it was found that the parameterized skip connections produced a more stable forecast with much lower variance than taking the output directly form the last layer. This approach also permitted targeting the output of each dilated convolutional layer with different levels of dropout [34] to ensure that the latent representations developed in the dilated convolutional layers were not overly influenced by past trends that may not be relevant in the recent forecast horizon of the time series. This technique was also used because the dataset contained a mixture of lengths and it was found that forecasts could be produced on the shorter length time series by penalizing the last two layers with 80% dropout instead of creating a separate model with a smaller receptive field for time series with shorter lengths. A residual connection was added from the input to the output of each layer to increase the training efficiency throughout the

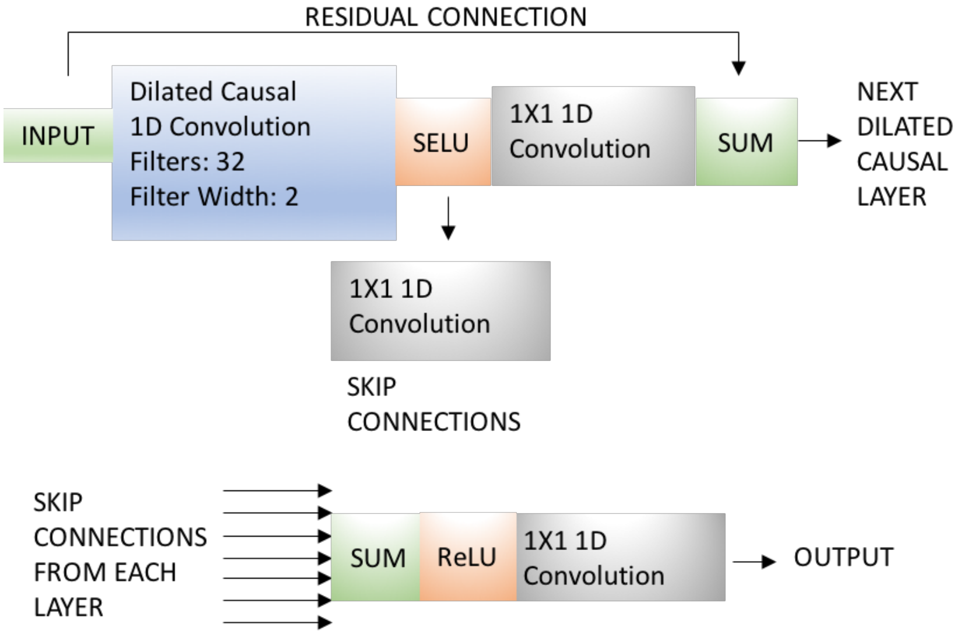

Figure 5: SeriesNet architecture

layers. The convolutional layer biases were not incorporated based on tests that determined including the biases resulted in worse performance.

# 5 Experiments

The experiments with SeriesNet were performed on the CIF2016 competition dataset [35]. The CIF2016 dataset was used to benchmark the performance of SeriesNet against the published results from different state of the art forecast models that were submitted to the competition [19]. In the experiments conducted, the CIF2016 dataset was not preprocessed with any re-scaling or transformations. The CIF2016 dataset summarized in Table 1 contained artificially generated and real datasets of different lengths and forecast horizons.

| Dataset | Total Series | Real Series | Generated Series | Period | Series Lengths | Forecast Horizons |
|---|---|---|---|---|---|---|
| CIF2016 | 72 | 24 | 48 | monthly | 23 to 108 months | 6 or 12 months |

Table 1: Summary of the CIF2016 dataset

The training of a dilated causal convolutional neural network has the advantage of being performed in parallel since the labels (i.e. next time step) for all inputs are known. This resulted in efficient training and fast convergence for SeriesNet with longest time series tested (i.e. 108 time steps) training in about 1 minute for 3000 epochs which was determined to optimize forecast performance on a validation set created from the input time series. The predictions of the SeriesNet were performed sequentially with the first prediction feedback into the model to generate the next prediction until the selected prediction horizon was achieved. The loss function of SeriesNet was chosen to be the mean absolute error (MAE) which was also utilized for the Augmented WaveNet. Training SeriesNet with a mean squared error (MSE) loss function did not converge. L2 regularization for the layer weights was selected from [0, 0.001, 0.005]. It was found that L2 weight regularization of 0.001 for all layers produced the best results. The weights of all layers were initialized with a truncated normal distribution with zero mean and a constant variance of 0.05. Xavier initialization [36] and He initialization [37] were tested but did not result in stable convergence during training. The Adam optimizer [38] was used with the learning rate selected from [2e-3, 1e-3, 7.5e-4] and $\beta_1$ from [0.5, 0.9]. It was found that a learning rate of 7.5e-4 and $\beta_1$ of 0.9 balanced training time and validation accuracy.

After tuning the model parameters during validation and training the network, the forecasts from SeriesNet were computed for the required forecast horizon and compared against the actual forecast values obtained from CIF2016 for each time series. The overall results from SeriesNet were averaged across 10 prediction trials. The competition submissions were evaluated on the symmetric mean absolute percentage error (SMAPE) of the forecast values (Ft) against the actual values (At) over the forecast horizon (n) as defined below in (2). The forecasts of the top 4 overall competition submissions and the baseline LSTM submission were compared to the forecasts from SeriesNet as shown in Table 2. The forecasts on the artificially generated series and the real series were also separated to compare differences. In Table 2, the numbers in brackets beside the forecasts indicate the overall rank of the submission in each category. The results of the CIF2016 competition were described in [19].

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{(|F_t| + |A_t|)/2} \tag{2}$$

| Model | Overall SMAPE | Generated Series SMAPE | Real Series SMAPE |
|---|---|---|---|
| LSTM-DES | 0.105 ± 0.107 (1) | 0.074 ± 0.053 (4) | 0.168 ± 0.152 (1) |
| LSTM-ES | 0.108 ± 0.116 (2) | 0.070 ± 0.051 (2) | 0.184 ± 0.165 (2) |
| *SeriesNet* | *0.117 ± 0.0003* | *0.0878 ± 0.0002* | *0.175 ± 0.001* |
| ES | 0.119 ± 0.142 (3) | 0.071 ± 0.057 (3) | 0.215 ± 0.202 (6) |
| MLP | 0.121 ± 0.135 (4) | 0.077 ± 0.060 (7) | 0.210 ± 0.191 (3) |
| LSTM | 0.133 ± 0.155 (10) | 0.080 ± 0.051 (9) | 0.241 ± 0.226 (13) |

Table 2: Summary of top CIF2016 results and SeriesNet results

The overall performance of SeriesNet was third compared to the ranking of the CIF2016 competition submissions. It compared well to the top 2 ensemble LSTM submissions that incorporated different models for the different forecast horizons and data preprocessing techniques such as removing seasonality from the data during model training. As discussed in Section 2, baseline LSTM models do not necessarily provide high forecast performance on time series data. The baseline LSTM submission shown in Table 2, ranked 10th overall in the competition. The CIF2016 competition results showed that deep learning techniques such as LSTM and multilayer perceptrons (shallow neural network) were not as effective on the generated datasets. Exponential smoothing (ES) models including a competition benchmark ES model and LSTM-ES were the most effective on the generated datasets which included datasets created from 4 types of trend (linear, exponential, saturated, and linear with break) with 2 types of random error and 2 types of seasonality. The performance of SeriesNet on the generated data sets was comparable to the baseline LSTM model and the benchmark autoregressive integrated moving average (ARIMA) model. The deep learning models including SeriesNet demonstrated the lowest SMAPE on the real data sets compared to the other models. The forecasts of the baseline LSTM model on the real data sets were improved by using data preprocessing to decompose the input data into trend, remainder and season. The LSTM model was trained on the trend and reminder, and the seasonality was added back to the predictions from the LSTM model. SeriesNet achieved comparable results to the LSTM-DES and LSTM-ES models on the real data sets without data preprocessing and less variability. As expected, 12 month forecasts from SeriesNet were worse than 6 month forecasts on the real time series. Examples of forecasts from SeriesNet on selected CIF2016 generated and real time series are displayed in the Appendix.

# 6 Limitations

This paper only explored the performance of the SeriesNet models on the CIF2016 dataset. Further analysis on different types of datasets is required to examine the capability of SeriesNet to forecast from different data distributions for varying forecast horizons. The role of data preprocessing, feature selection and different receptive fields on the forecast performance of SeriesNet was not explored in this paper. As demonstrated in the CIF2016 winning LSTM submission, data preprocessing had a significant effect on the forecast results. The comparisons made in this paper were limited only to the results from models submitted to the CIF2016 competition. Other approaches to time series forecasting such as Bayesian methods, Gaussian processes or ensemble methods were not compared to further examine the benefits and trade-offs from the different approaches. For example, it was found that on the CIF2016 generated series that the deep learning forecasting approaches were not necessarily more effective than exponential smoothing approaches. Also forecasting metrics other than SMAPE were not compared to which may indicate different aspects of the forecast quality from SeriesNet. The SeriesNet implementation in this paper tested different model architecture components and hyperparameter configurations for the CIF2016 dataset although these settings were not exhaustively tested. It was found that the forecasts were very sensitive to layer weight initialization, receptive field, training duration and levels of dropout. Forecast improvements may have been demonstrated by creating an ensemble of SeriesNet models to account for the different time series input lengths and forecast horizons but this was not tested. Therefore, it is not clear whether the implementation in this paper was optimal and whether it would generalize well on other datasets. Further work is required to further understand how the model architecture and hyperparameters impact the forecast capability of SeriesNet in different contexts and forecast horizons.

# 7 Conclusions

In this paper, deep learning techniques for time series forecasting were surveyed and it was demonstrated that SeriesNet, a dilated causal convolutional neural network implementation was effective in producing overall forecasts comparable to the winning LSTM submission in the CIF2016 competition [19] without ensemble methods or data preprocessing. Although SeriesNet performance was not as effective on the CIF2016 generated time series as the winning LSTM submission. The implementation created in this paper was inspired by the original WaveNet architecture [17] and the Augmented WaveNet [26] created for time series forecasting. This paper extended the work of [26] by introducing the SELU activation function in the dilated causal convolutions layers and outputting the layer activations via parameterized skip connections to the output. This design demonstrated that the different scales of the input time series did not adversely impact the forecasts and therefore no data preprocessing was required to obtain comparable forecasts to other techniques. Also the use of the parameterized skip connections from each dilated causal convolutional layer to the output produced stable forecasts that did not significantly vary when conducting multiple trials. Larger variation was observed for the Augmented WaveNet direct output design. It was also shown that dropout was effective in targeting the dilated causal convolutions layers with longer dilations to reduce their contribution on the forecasts since the past information in the time series may not be as relevant to the recent progression.

Areas for further research include tuning techniques for the SeriesNet architecture and hyperparameters to find optimal configurations for different inputs and forecast horizons. Different filters can be investigated to condition or weight recent time lags in the input. This may be more informative for forecasting similar to attention models [39] used in language translation. The use of different non-linear embeddings of the input time series to extract more representative features may improve SeriesNet forecasts, as it was demonstrated in [20] and [22] that for LSTM forecasting, results were improved by encoding the inputs before training the model. The work in [40] demonstrated that an embedded input could be combined with a dilated convolutional neural net to perform predictions for text modelling. Recent work on learning theory for forecasting time series [41] may also provide insights for augmenting SeriesNet with features to identify or correct for deviations from non-stationarity which may lead to improved forecasting. The analysis provided in [41] may also be further explored to determine what guarantees and stability bounds can be determined for forecasts from dilated causal convolution neural networks.

# References

[1] Jenkins Box. *Time Series Analysis Forecasting and Control.* 4 edition, 2013.

[2] Thomas Dietterich. Machine learning for sequential data: A review. *Structural, syntactic, and statistical pattern recognition*, pages 227–246, 2002.

[3] Nesreen K. Ahmed, Amir F. Atiya, Neamat El Gayar, and Hisham El-Shishiny. An Empirical Comparison of Machine Learning Models for Time Series Forecasting. *Econometric Reviews*, 29(5-6):594–621, August 2010.

[4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, May 2015.

[5] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29(6):82–97, November 2012.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May 2017.

[7] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv:1602.07261 [cs]*, February 2016.

[8] John Cristian Borges Gamboa. Deep Learning for Time-Series Analysis. *arXiv:1701.01887*, 2017.

[9] M. Ghiassi, H. Saidane, and D.K. Zimbra. A dynamic artificial neural network model for forecasting time series events. *International Journal of Forecasting*, 21(2):341–362, April 2005.

[10] R. J. Frank, N. Davey, and S. P. Hunt. Time Series Prediction and Neural Networks. *Journal of Intelligent and Robotic Systems*, 31(1-3):91–103, May 2001.

[11] Jim Gao. Machine Learning Applications for Data Center Optimization. 2014.

[12] Olalekan Ogunmolu, Xuejun Gu, Steve Jiang, and Nicholas Gans. Nonlinear Systems Identification Using Deep Dynamic Neural Networks. *arXiv:1610.01439*, 2016.

[13] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv:1611.03530 [cs]*, November 2016.

[14] S. F. Crone and S. Häger. Feature selection of autoregressive Neural Network inputs for trend Time Series Forecasting. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1515–1522, July 2016.

[15] John Alberg and Zachary C. Lipton. Improving Factor-Based Quantitative Investing by Forecasting Company Fundamentals. *arXiv:1711.04837*, 2017.

[16] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, and Klaus Macherey. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144*, 2016.

[17] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv:1609.03499*, 2016.

[18] Slawek Smyl and Karthik Kuber. Data Preprocessing and Augmentation for Multiple Short Time Series Forecasting with Recurrent Neural Networks. July 2016.

[19] M. tpnika and M. Burda. On the results and observations of the time series forecasting competition CIF 2016. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6, July 2017.

[20] Valentin Flunkert, David Salinas, and Jan Gasthaus. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. *arXiv:1704.04110 [cs, stat]*, April 2017.

[21] Kasun Bandara, Christoph Bergmeir, and Slawek Smyl. Forecasting Across Time Series Databases using Recurrent Neural Networks on Groups of Similar Series: A Clustering Approach. *arXiv:1710.03222 [cs, econ, stat]*, October 2017.

[22] Nikolay Laptev, Jason Yosinski, Li Erran Li, and Slawek Smyl. Time-series extreme event forecasting with neural networks at uber. In *International Conference on Machine Learning*, 2017.

[23] Roni Mittelman. Time-series modeling with undecimated fully convolutional neural networks. *arXiv:1508.00317*, 2015.

[24] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 1578–1585. IEEE, 2017.

[25] Anastasia Borovykh, Sander Bohte, and Cornelis W. Oosterlee. Conditional Time Series Forecasting with Convolutional Neural Networks. *arXiv:1703.04691v1*, 2017.

[26] Anastasia Borovykh, Sander Bohte, and Cornelis W. Oosterlee. Conditional Time Series Forecasting with Convolutional Neural Networks. *arXiv:1703.04691v3*, 2017.

[27] Subin Yi, Janghoon Ju, Man-Ki Yoon, and Jaesik Choi. Grouped Convolutional Neural Networks for Multivariate Time Series. *arXiv:1703.09938*, 2017.

[28] Mikolaj Binkowski, Gautier Marti, and Philippe Donnat. Autoregressive Convolutional Neural Networks for Asynchronous Time Series. *arXiv:1703.04122 [cs]*, March 2017.

[29] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional Image Generation with PixelCNN Decoders. *arXiv:1606.05328 [cs]*, June 2016.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. *arXiv:1603.05027 [cs]*, March 2016.

[31] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *PMLR*, pages 448–456, June 2015.

[32] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[33] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-Normalizing Neural Networks. *arXiv:1706.02515 [cs, stat]*, June 2017.

[34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[35] http://irafm.osu.cz/cif/main.php. CIF - Computational Intelligence in Forecasting.

[36] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, March 2010.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv:1502.01852 [cs]*, February 2015.

[38] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, December 2014.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv:1706.03762 [cs]*, June 2017.

[40] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved Variational Autoencoders for Text Modeling using Dilated Convolutions. *arXiv:1702.08139 [cs]*, February 2017.

[41] Vitaly Kuznetsov and Mehryar Mohri. Theory and Algorithms for Forecasting Time Series. *arXiv:1803.05814 [cs]*, March 2018.

# Appendix

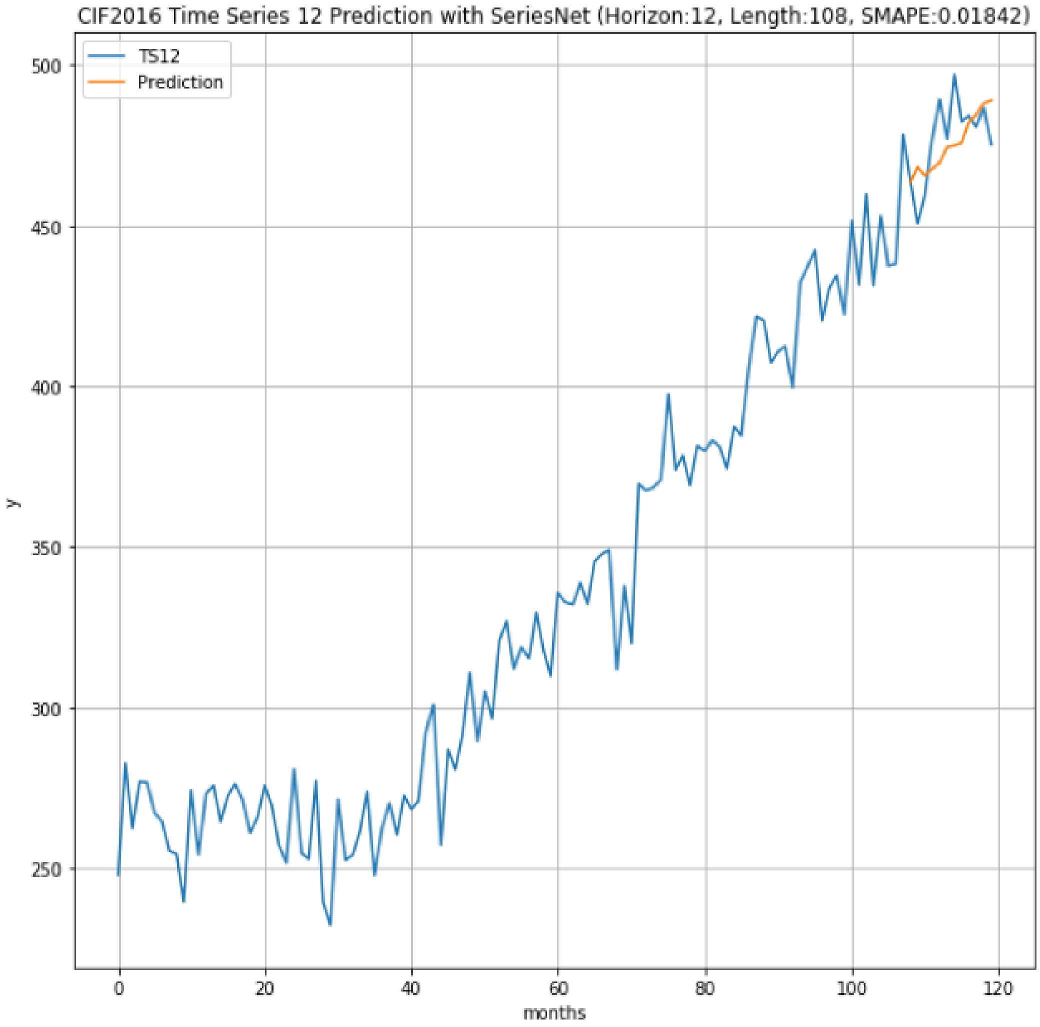

Figure 6: CIF2016 Generated Series 12

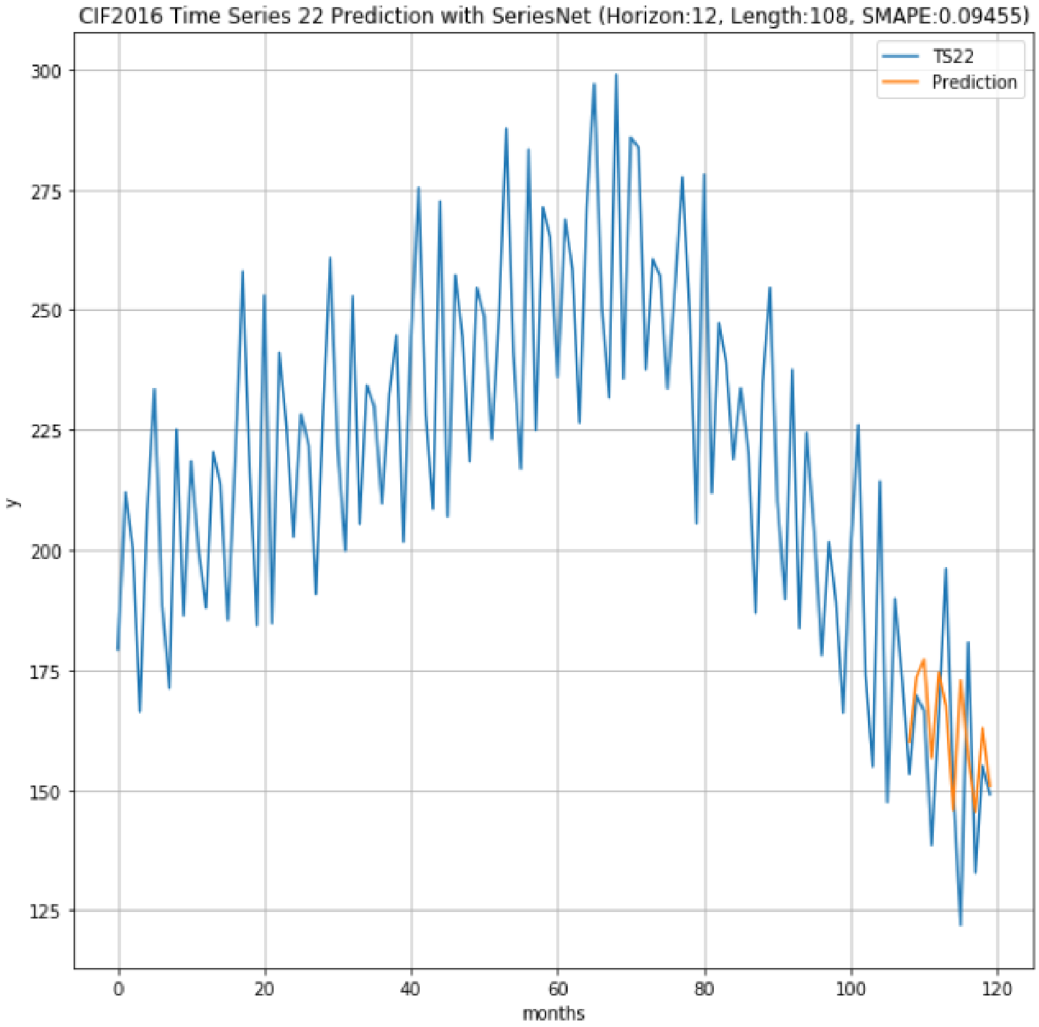

Figure 7: CIF2016 Generated Series 22

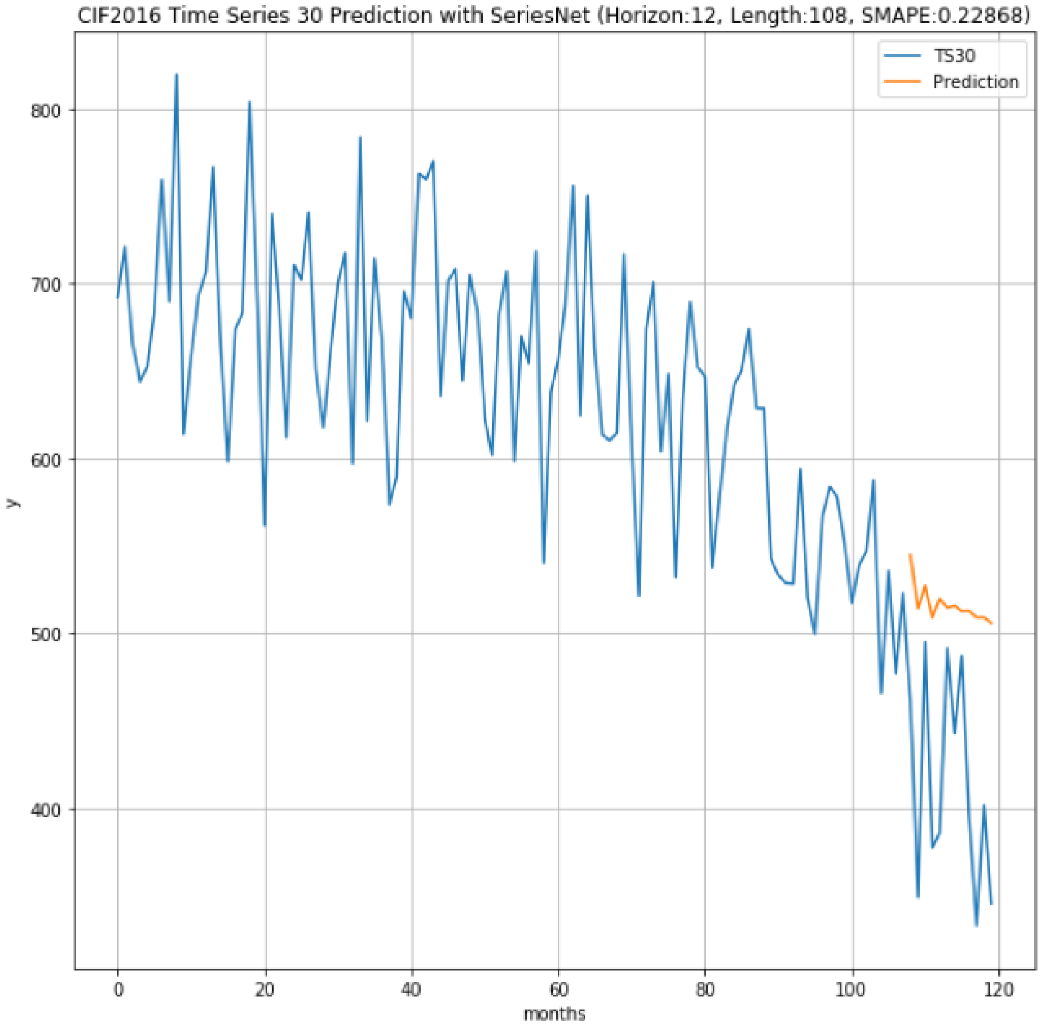

Figure 8: CIF2016 Generated Series 30

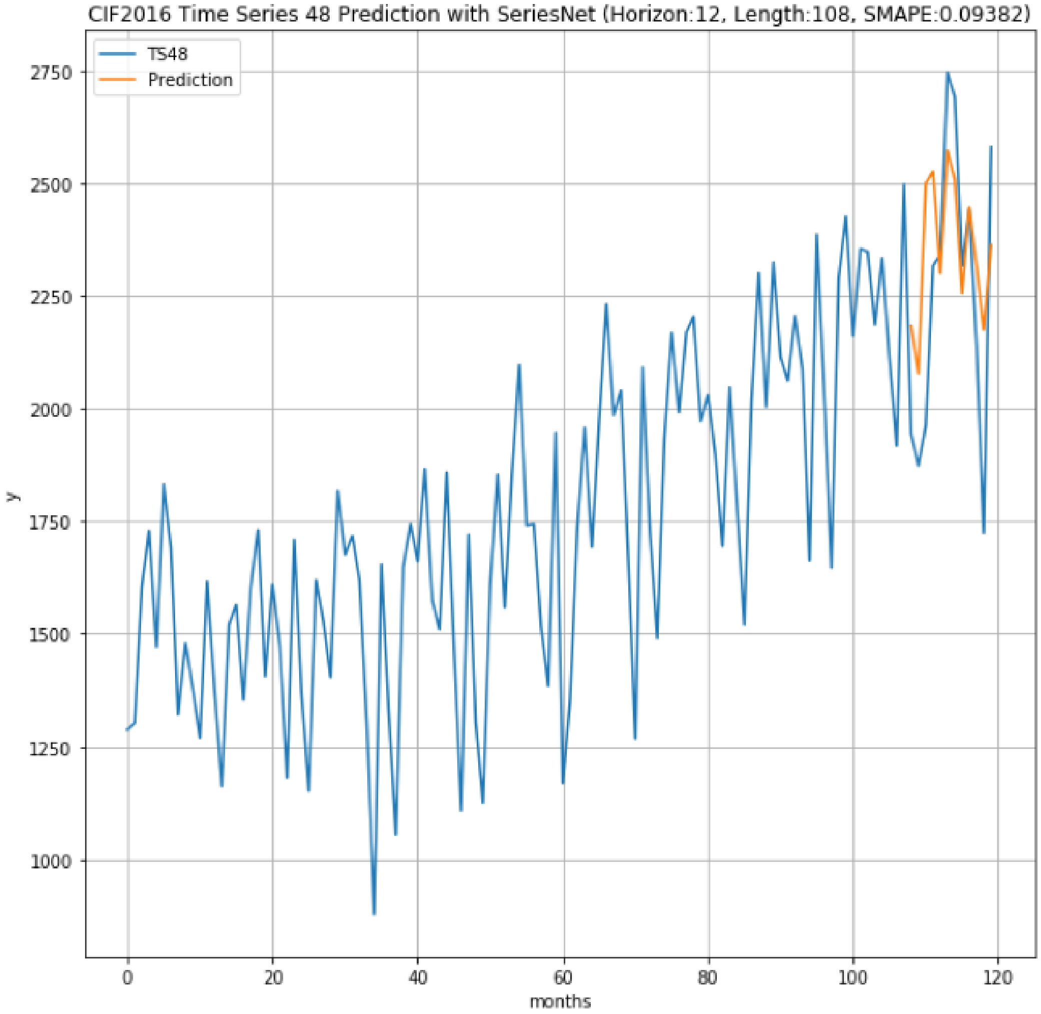

Figure 9: CIF2016 Generated Series 48

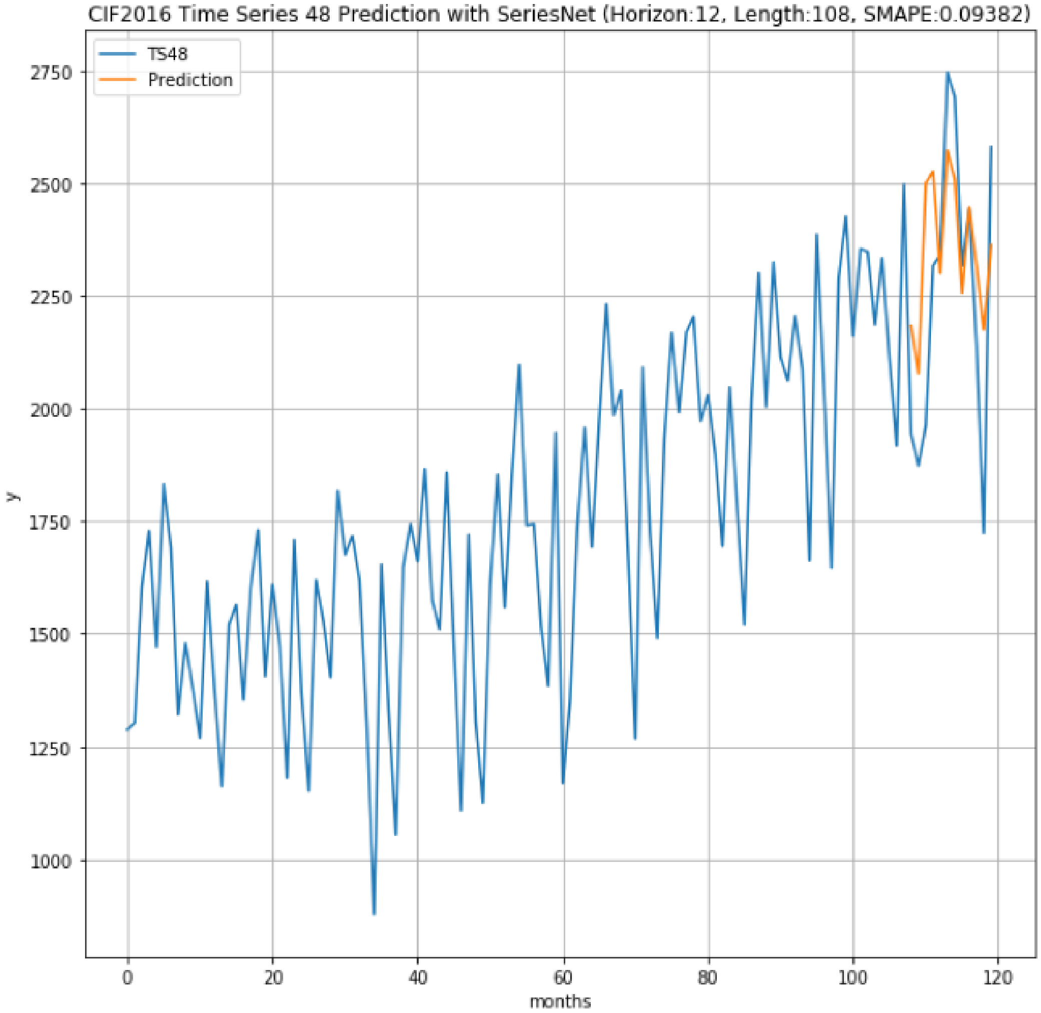

Figure 10: CIF2016 Real Series 51

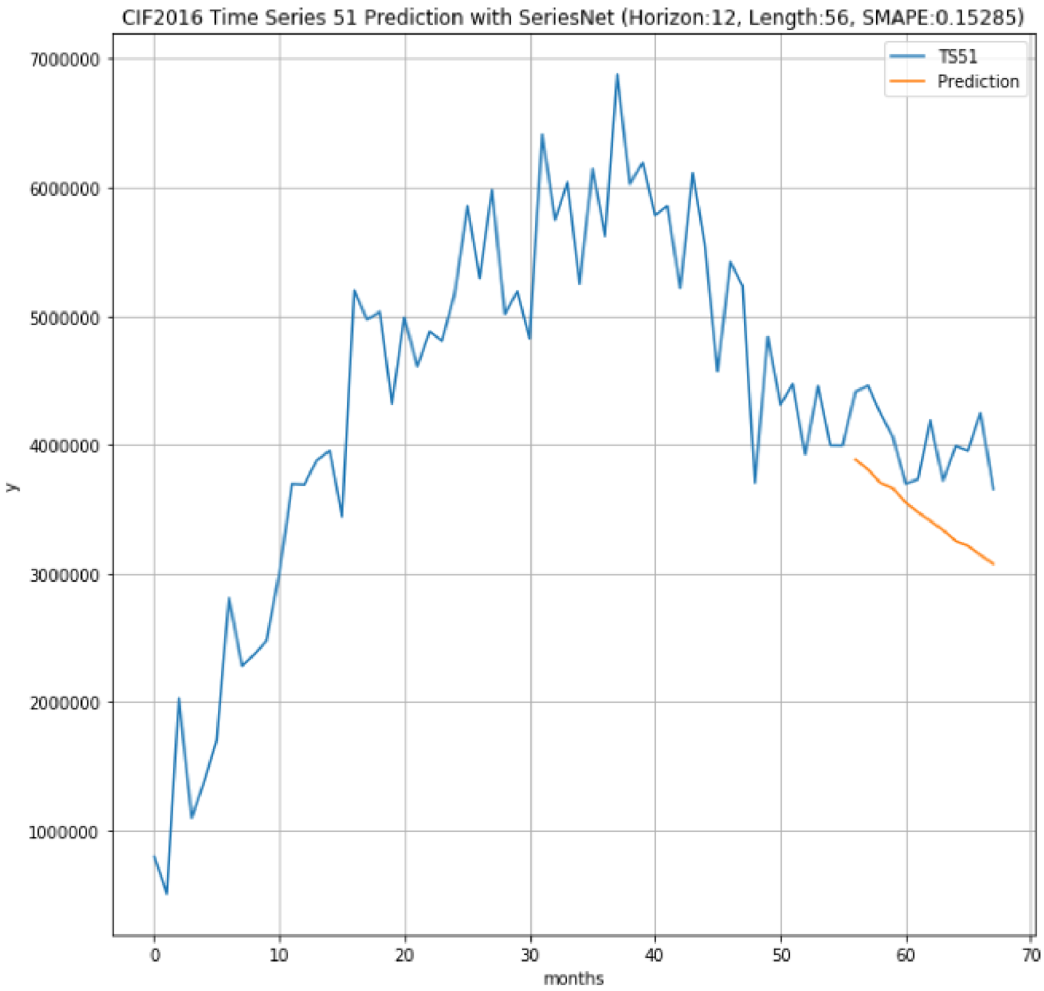

Figure 11: CIF2016 Real Series 56

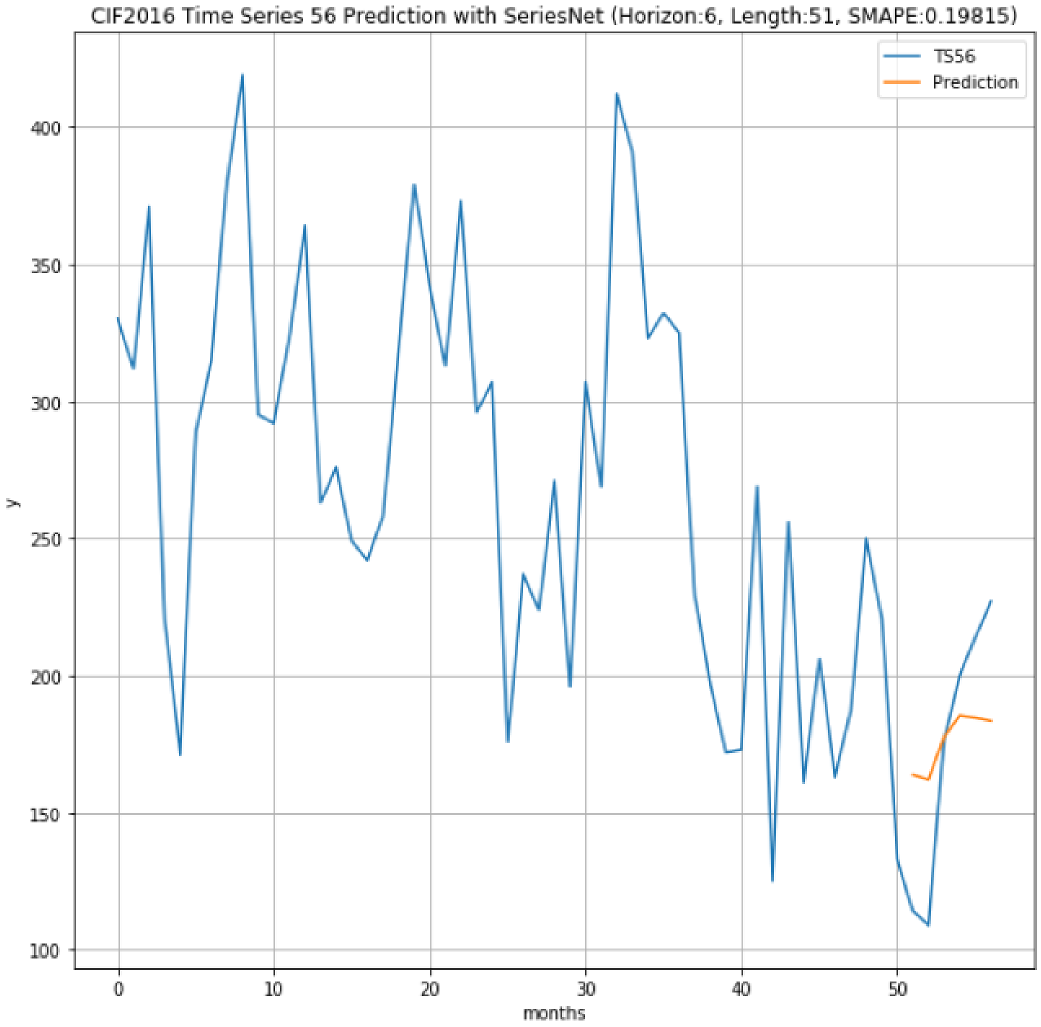

Figure 12: CIF2016 Real Series 56

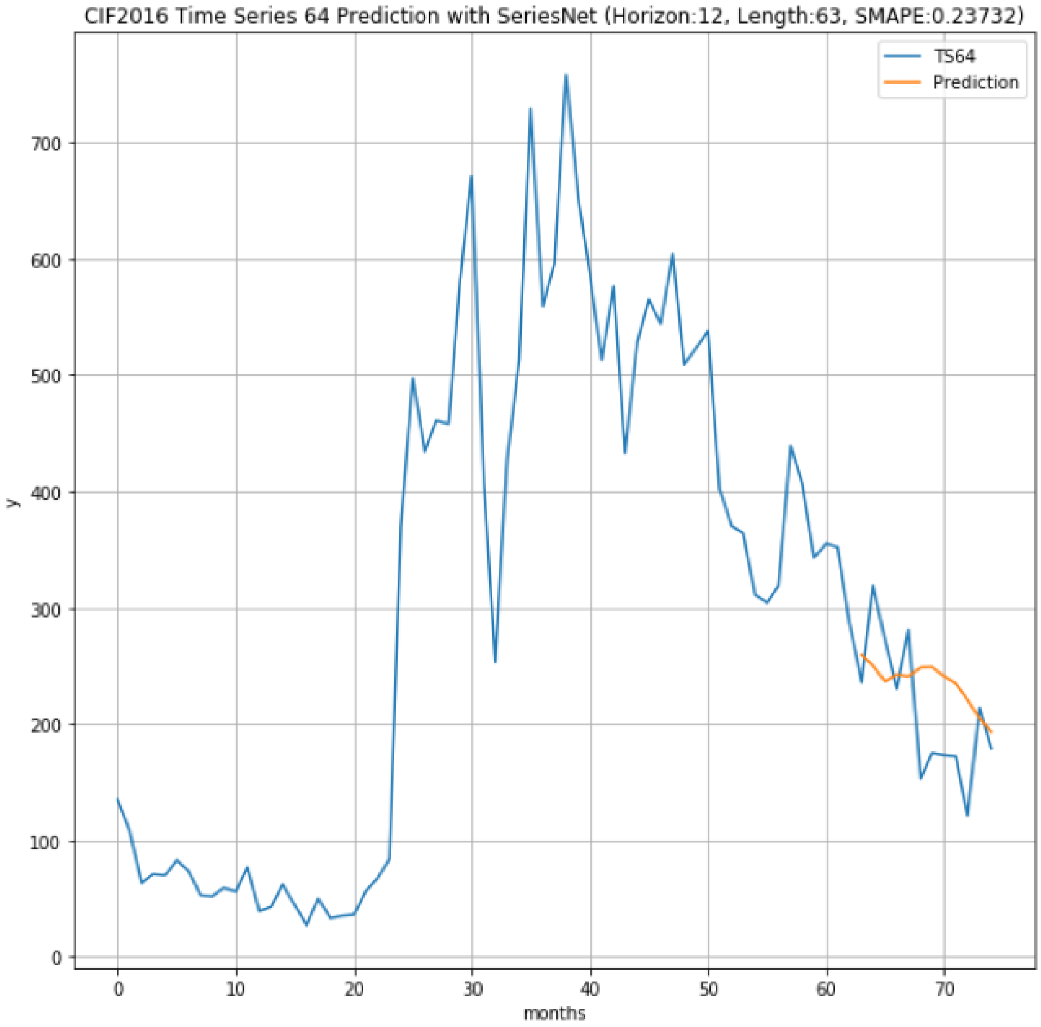

Figure 13: CIF2016 Real Series 64

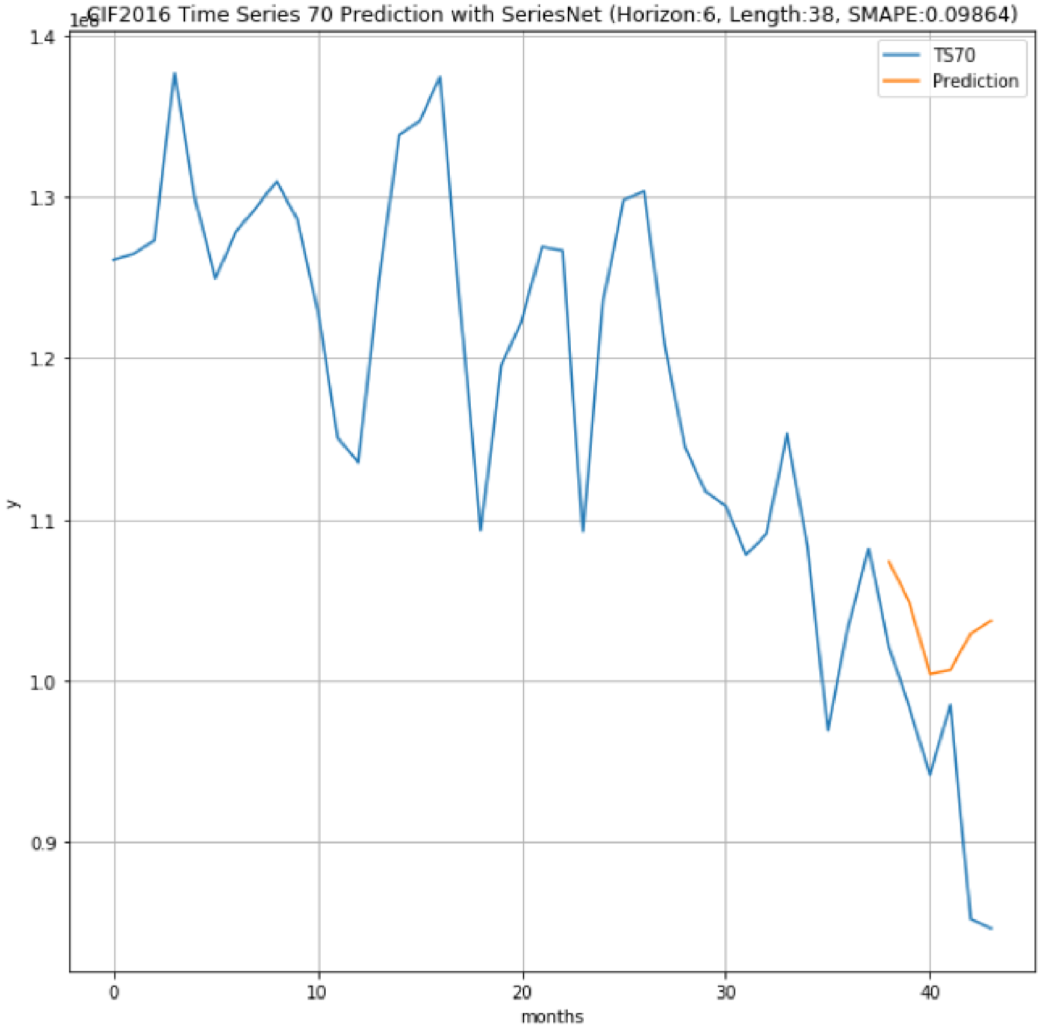

Figure 14: CIF2016 Real Series 70